

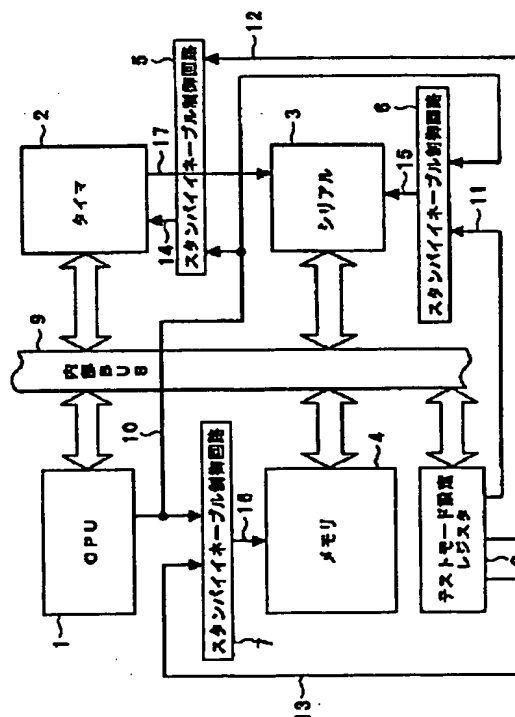
(2) Japanese Patent Application Laid-Open No.2001-202350

“Standby Control Device of Microcomputer”

The following is an extract relevant to the present application.

5

The present invention relates to a standby control device which is configured so that whether or not built-in peripheral units escape from a standby state when a standby mode is released can be controlled at a time of a test.



## 【特許請求の範囲】

【請求項 1】 CPU と、複数の機能ユニットとを備えた半導体集積回路装置において、

前記 CPU は、前記複数の機能ユニットに対して、スタンバイ状態の設定及び解除を制御するスタンバイ信号を出力し、

前記 CPU からスタンバイ解除が指示された時に、前記複数の機能ユニットに対してそれぞれ個別にスタンバイ状態をそのまま維持するか解除するかを制御するための制御信号を出力する手段と、

前記制御信号と、前記 CPU から出力されるスタンバイ信号とを入力し、前記複数の機能ユニットのそれぞれに対してスタンバイ状態の設定及び解除を制御する手段と、

を備えたことを特徴とする半導体集積装置。

【請求項 2】 CPU と、複数の周辺ユニットとを備えたマイクロコンピュータのスタンバイ制御装置において、前記周辺ユニットのそれぞれについてスタンバイ状態を維持するか解除するかを指定する制御信号を出力するテストモード設定レジスタと、

前記テストモード設定レジスタからの制御信号と、前記 CPU から出力されるスタンバイ信号とを入力し、前記制御信号と前記スタンバイ信号の値に基づき、前記周辺ユニットのそれぞれに対してスタンバイ状態の設定及び解除を制御する信号を出力するスタンバイイネーブル制御回路と、

を備えたことを特徴とするマイクロコンピュータのスタンバイ制御装置。

【請求項 3】 テスト実行時、一旦条件を設定した後、次のテストにおいて条件を変更する必要がある周辺ユニットについてはスタンバイ状態をそのまま維持し、条件の変更を行う周辺ユニットのみスタンバイ状態を解除して、新たな条件を設定した後、再度スタンバイ状態に設定して次のテストが行われる、ことを特徴とする、請求項 2 記載のマイクロコンピュータのスタンバイ制御装置。

【請求項 4】 前記テストモード設定レジスタには、前記 CPU から制御内容が設定され、前記テストモード設定レジスタは、前記 CPU から受け取った制御内容に基づき、前記各周辺ユニットに対して、スタンバイ状態を維持するか解除するかを指定する制御信号を出力する、ことを特徴とする、請求項 2 記載のマイクロコンピュータのスタンバイ制御装置。

【請求項 5】 前記テストモード設定レジスタが、外部から入力される外部制御信号に基づき、前記各周辺ユニットに対して、スタンバイ状態を維持するか解除するかを指定する制御信号を出力する、ことを特徴とする、請求項 2 記載のマイクロコンピュータのスタンバイ制御装置。

【請求項 6】 CPU と、前記 CPU とバスを介して接続

されるメモリ及び／又は周辺回路を含む複数の周辺ユニットと、を備えたマイクロコンピュータにおいて、

前記 CPU から出力される信号を入力とし、前記複数の周辺ユニットについてスタンバイ状態を維持するか解除するかを指定する制御信号をそれぞれ出力するテストモード設定レジスタと、

前記周辺ユニット毎に設けられ、前記テストモード設定レジスタから出力される制御信号と、前記 CPU から出力される前記複数の周辺ユニットに共通のスタンバイ信号と、を入力とし、前記制御信号がインアクティブ状態

とされ、前記スタンバイ信号がアクティブ状態からインアクティブ状態とされた場合、前記周辺ユニットのスタンバイ状態を解除し、前記制御信号がアクティブ状態のときには、前記スタンバイ信号がアクティブ状態からインアクティブ状態とされた場合にも、前記周辺ユニットのスタンバイ状態を維持するように制御する信号を、対応する前記周辺ユニットに対してそれぞれ出力する複数のスタンバイイネーブル制御回路と、

を備えたことを特徴とするマイクロコンピュータ。

【請求項 7】 CPU と、前記 CPU とバスを介して接続されるメモリ及び／又は周辺回路を含む複数の周辺ユニットと、を備えたマイクロコンピュータにおいて、外部から入力される外部制御信号を入力とし、前記複数の周辺ユニットについてスタンバイ状態を維持するか解除するかを指定する制御信号をそれぞれ出力するテストモード設定レジスタと、

前記周辺ユニット毎に設けられ、前記テストモード設定レジスタから出力される制御信号と、前記 CPU から出力される前記複数の周辺ユニットに共通のスタンバイ信号と、を入力とし、前記制御信号がインアクティブ状態とされ、前記スタンバイ信号がアクティブ状態からインアクティブ状態とされた場合、前記周辺ユニットのスタンバイ状態を解除し、前記制御信号がアクティブ状態のときには、前記スタンバイ信号がアクティブ状態からインアクティブ状態とされた場合にも、前記周辺ユニットのスタンバイ状態を維持するように制御する信号を、対応する前記周辺ユニットに対してそれぞれ出力する複数のスタンバイイネーブル制御回路と、

を備えたことを特徴とするマイクロコンピュータ。

【請求項 8】 1 又は複数の前記周辺ユニットをスタンバイ状態として一のテストが実行され、次のテストの実行にあたり、条件の設定変更を行う周辺ユニットに対してのみ、前記周辺ユニットのスタンバイ状態を解除してあたかな条件に設定した後スタンバイ状態に設定し、条件設定の変更を行わない他の周辺ユニットは、この間、スタンバイ状態をそのまま維持するように、制御が行われる、ことを特徴とする請求項 6 又は 7 記載のマイクロコンピュータ。

【請求項 9】 前記テストが、静止状態の電源電流を測定する IDDQ テストである、ことを特徴とする請求項 8

記載のマイクロコンピュータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はマイクロコンピュータのスタンバイ制御装置に関し、特に、IDDQテストの実行に好適なマイクロコンピュータのスタンバイ制御装置に関する。

【0002】

【従来の技術】LSI（大規模集積回路）のテストを行う方法として、スタンバイ状態のとき流れる電流を測定して、良、不良を判定するテスト（以下「IDDQテスト」という）は、不良検出に有効なテスト手法であり、スタンバイ電流を、例えばLSIの内部状態を変えながら測定することが有効であることが知られている。すなわち、マイクロコンピュータのIDDQテストでは、対象となる周辺ユニットやメモリ、CPUの設定条件を変えながらテストを実行する。

【0003】CMOS構成のLSIでは、スタンバイ状態時に流れる電流は、例えばマイクロアンペアオーダーとされており、不良のトランジスタが存在した場合に流れる静止時の電源電流（VDD supply current（ $I_{ss}$ ）

Quiescent）の電流値は、この正常時の電流値を大きく上回ることになり、IDDQテストで不良が検出される。

【0004】そして、IDDQテストを実行する場合、CPU（中央演算処理装置）のプログラム命令で内蔵する周辺ユニットの状態をテスト毎に設定する。マイクロコンピュータが内蔵する周辺ユニットが複数存在するときは、各々の周辺ユニットに状態を設定してスタンバイ命令により、スタンバイ状態へ移行する。

【0005】ところで、周辺ユニットにおいて、スタンバイ条件が適切に設定がされていない場合には、スタンバイ状態でも電源電流が流れる場合がある。このため、周辺ユニットに対するスタンバイ条件の設定を省略することはできない。

【0006】図4は、従来のマイクロコンピュータのスタンバイ制御装置の構成の一例を示す図である。図5は、図4に示した従来のマイクロコンピュータで、IDDQテストを行った場合のプログラムの流れを示したフローチャートである。

【0007】図4を参照すると、従来のマイクロコンピュータのスタンバイ制御装置は、CPU1と、タイマ2と、シリアルインタフェース（シリアル）3と、CPU1から出力され、メモリ4とタイマ2とシリアルインタフェース3の各周辺ユニットへ共通に入力されるスタンバイ信号10と、タイマ2から出力されシリアルインタフェース3のシリアルクロックの基準となるタイミング信号17と、CPU1と、メモリ4と、タイマ2と、シリアルインタフェース3等の各周辺ユニットを接続する内部バス（BUS）9を備えて構成されている。

【0008】図4及び図5を参照して、この従来のマイクロコンピュータにおいてIDDQテストを行った場合の処理手順について説明する。

【0009】まず、CPU1はタイマ2に条件1、シリアルインタフェース3に条件1、メモリ4に条件1を設定する（ステップ501～503）。つづいてCPU1はスタンバイ命令を実行し（ステップ504）、スタンバイ制御信号10がアクティブになる。またタイマ2から出力するタイミング信号17はアクティブになる。この条件で、スタンバイ電流を測定する（ステップ505）。

【0010】スタンバイ状態が解除されると（ステップ506）、各周辺ユニット全て通常動作へ移行する。

【0011】次に、CPU1はタイマ2に条件1、シリアルインタフェース3に条件1、メモリ4に条件2を設定する（ステップ507～509）。つづいてCPU1はスタンバイ命令を実行し（ステップ510）、スタンバイ制御信号10がアクティブになる。この条件で、スタンバイ電流を測定する（ステップ511）。

【0012】スタンバイ状態が解除されると（ステップ512）、各周辺ユニット全て通常動作へ移行する。

【0013】次に、CPU1はタイマ2に条件1、シリアルインタフェース3に条件2、メモリ4に条件1を設定する（ステップ513～515）。つづいてCPU1はスタンバイ命令を実行し（ステップ516）、スタンバイ制御信号10がアクティブになる。この条件でスタンバイ電流を測定する（ステップ517）。

【0014】スタンバイ状態が解除されると（ステップ518）、各周辺ユニット全て通常動作へ移行する。

【0015】次に、CPU1はタイマ2に条件2、シリアルインタフェース3に条件1、メモリ4に条件1を設定する（ステップ519～521）。つづいてCPU1はスタンバイ命令を実行し（ステップ522）、スタンバイ制御信号10がアクティブになる。またタイマ2から出力するタイミング信号17はインアクティブになる。この条件でスタンバイ電流を測定する（ステップ523）。

【0016】スタンバイ状態が解除されると（ステップ524）、各周辺ユニット全て通常動作へ移行する。

【0017】次に、CPU1はタイマ2に条件2、シリアルインタフェース3に条件2、メモリ4に条件1を設定する（ステップ525～527）。次にCPU1はスタンバイ命令を実行し（ステップ528）、スタンバイ制御信号10がアクティブになる。この条件でスタンバイ電流を測定する（ステップ529）。

【0018】スタンバイ状態が解除されると（ステップ530）、各周辺ユニット全て通常動作へ移行する。

【0019】このように、図4に示した従来のマイクロコンピュータにおいては、CPU1からのスタンバイ信号10をインアクティブとしてスタンバイ状態から解除

すると、内蔵される全ての周辺ユニットが通常動作を開始するため、再度条件を設定する必要があった。

【0020】すなわち、従来のマイクロコンピュータにおいては、スタンバイ条件を、測定毎に設定して、スタンバイ状態に設定して電流（電源電流）を測定していたため、IDDQテストプログラム（図5では30のステップ）が膨大なものとなり、テストコストが高くなる。また膨大なテストパターンを収納することができる高価なLSIテストが必要とされる。

【0021】なお、例えば特開平10-283275号公報には、内蔵する周辺ユニット毎にスタンバイ状態を設定できる集積回路の構成が開示されている。図6は、この従来の集積回路の構成を示す図である。スイッチング回路613の複数の電力制御スイッチ639～642が使用されて複数の外部電源信号のうちそれが各メモリブロック626、628、630、632、634に供給するかを動的に選択する。電力制御スイッチは、レジスタ24にデータを書き込むことで、ソフトウェアで構成でき、テスト制御回路22によりイネーブルにすることができ、またVDD電源障害に応じて自動的にイネーブルされるように構成され、インテリジェントコントローラがデータアクセスの実行フロー及びメモリバンクからの命令フェッチに応じて動的にスイッチを制御でき、現在アクセスされているメモリバンクまたは最近アクセスされたメモリバンクのみ高電力レベルに活性化され、他の全てのメモリバンクは低電力スタンバイモードにできるように構成され、複数のメモリバンクの間で一組のメモリバンクを集積回路内のスタンバイメモリとして動的に選択可能とした構成が開示されている。メモリチップにおいて複数のメモリマクロを搭載している場合に、読み出しを行うメモリマクロやその前後のメモリマクロを通常のアクセス状態にし、その他のメモリマクロをスタンバイ状態にする。さらに未使用メモリマクロに供給する電源電圧を落としている。

【0022】複数のメモリマクロを搭載するメモリでは、アクセスされるメモリマクロとアクセスされないメモリマクロがある。アクセスされないメモリマクロに対しても、制御信号をアクティブにして、電源電圧を加えると電力を消費する。この従来のメモリでは、アクセスされていないメモリマクロをスタンバイ状態にして、かつ供給する電源電圧を低下させて余分に消費される電力を低下させている。

【0023】図6に示す構成において、この制御は、制御回路620内のレジスタ24とテストコントロール22によって行われている。

【0024】制御回路620から出力される信号により、スイッチ639、640、642を動作させ、スイッチから出力される信号VDARRにより、スタンバイ状態とするメモリマクロと、通常動作させるメモリマクロとを制御している。この従来の集積回路は、メモリチ

ップの消費電力を低減するためのものである。

【0025】1チップマイクロコンピュータと相違して、メモリは同一のメモリマクロを複数搭載する構成とされており、テストをする場合でも、同じプログラムを僅かな変更を行うだけで、メモリセルに値を容易に設定できる。

【0026】またメモリマクロはアクセスされないときには内部に保持している状態は変わらないので、スタンバイ状態を設定しやすく、スタンバイ状態を解除しても、アクセスされないメモリマクロの保持値が変化しない。

【0027】これに対して、1チップマイクロコンピュータでは、内蔵する周辺ユニットやCPUは内部クロックが変化する度に、内部に保持した値が変化する。このためIDDQテストを実行する時は、保持させる値をタイミング良く設定する必要がある。

【0028】また搭載する周辺ユニットは例えばタイマやシリアルインタフェース等のように互いに異なる機能を具備しているため、メモリのように同じメモリマクロを搭載する場合に比べ、容易に内部の条件を設定できない。

【0029】更に、マイクロコンピュータでは周辺ユニットどうしが制御信号で接続されている場合もあり、制御信号の状態を考慮する必要がある。

【0030】以上述べたように、特開平10-283275号公報等に記載される構成をマイクロコンピュータに適用しても、本来の目的であるIDDQテストを効率良く行うことはできない、という問題点を有している。

【0031】

【発明が解決しようとする課題】以上説明した、従来の技術の問題点について以下に要約して説明する。

【0032】上記した通り、図4に示した従来のマイクロコンピュータのスタンバイ制御装置は、スタンバイ状態を解除すると、CPUを初め周辺ユニットが一斉にスタンバイ状態から抜けて、通常動作に移行する。一旦、クロック信号が動作すると、CPUや周辺ユニットの内部状態が変化してしまう。

【0033】このため、スタンバイ時の電流を測定して、マイクロコンピュータの異常の検出を行うIDDQテストを実行しようとする、周辺ユニットの状態やCPUの状態を、その都度、プログラム命令を実行させて設定する必要があり、複数の条件を周辺ユニットに設定し、これを組み合わせてIDDQテストを実行する場合、必要とされるプログラム命令が複雑となり、かつ、プログラム命令のサイズも大きなものになる、という問題点を有している。その結果、テストコストを上昇させることになる。

【0034】また、図6に示したように、メモリで搭載するメモリマクロを任意にスタンバイ状態にできるようにした構成も知られているが、これは、アクセスされて

10

20

30

40

50

いないメモリマクロをスタンバイ状態にすることで消費電力を低減するためのものである。

【0035】メモリで、スタンバイ電流を測定することは、メモリセルに“1”または“0”を書き込んで行う単純なものであり、メモリの場合、スタンバイ電流を測定して不良を検査するIDDQテストでは、ALL1書き込み、ALL0書き込み、チェッカーデータ書き込み、チェッカーバーデータ書き込みの4通りで、基本的なテストが完了する。

【0036】しかしながら、マイクロコンピュータに内蔵される周辺ユニットは、ランダム回路よりなり、かつ、周辺ユニット同士が制御信号で接続されている場合もあり、制御信号のアクティブ、インアクティブの条件を考慮する必要があることから、メモリマクロのように、単純な設定では、IDDQテストを行うことができない、という問題点も有している。

【0037】したがって、本発明は、上記問題点に鑑みてなされたものであって、その目的は、マイクロコンピュータにおいて、IDDQテストのプログラム容量を削減し、実行するテストボタンを短縮可能とし、テストコストを低減する、スタンバイ制御装置を提供することにある。

【0038】

【課題を解決するための手段】前記目的を達成する本発明は、CPUと、複数の周辺ユニットとを備えたマイクロコンピュータのスタンバイ制御装置において、前記周辺ユニットのそれぞれについてスタンバイ状態を維持するか解除するかを指定する制御信号を出力するテストモード設定レジスタと、前記テストモード設定レジスタからの制御信号と、前記CPUから出力されるスタンバイ信号とを入力し、前記周辺ユニットのそれぞれに対してスタンバイ状態の設定及び解除を制御するスタンバイイネーブル制御回路と、を備えている。

【0039】

【発明の実施の形態】本発明の実施の形態について以下に説明する。本発明は、その好ましい一実施の形態において、CPU(1)と、複数の周辺ユニット(2、3、4)と、周辺ユニットのそれぞれについてスタンバイ状態を維持するか解除するかを指定する制御信号を出力するテストモード設定レジスタ(8)と、前記テストモード設定レジスタ(8)からの制御信号(12、11、13)と、CPU(1)から出力される複数の周辺ユニット(2、3、4)に共通のスタンバイ信号(10)を入力し、複数の周辺ユニット(2、3、4)のそれぞれに対してスタンバイ状態の設定及び解除を制御する信号(14、15、16)を出力するスタンバイイネーブル制御回路(5、6、7)と、を備えている。

【0040】本発明の一実施の形態においては、テスト時に、一旦条件を設定した後、次のテストにおいて変更する必要がない周辺ユニットのスタンバイ状態をそのま

ま維持可能とし、条件の変更を行う周辺ユニットのみスタンバイ状態を解除し、新たな条件を設定した後、スタンバイ状態に設定してテストを行う。

【0041】本発明の一実施の形態においては、テストモード設定レジスタ(8)が、CPU(1)から制御される。

【0042】本発明の一実施の形態においては、テストモード設定レジスタが、外部から入力される外部制御信号(18、19、20)に基づき、スタンバイ状態を維持するか解除するかを指定する制御信号(11、12、13)を出力する構成としてもよい。

【0043】本発明によれば、テストモード設定レジスタ(8)にCPUがスタンバイ解除の条件を書きこみ、各々の周辺ユニット(2、3、4)に付加されたスタンバイイネーブル制御回路(5、6、7)によってスタンバイ状態からの解除許可、禁止を行う構成としている。一旦、スタンバイ状態で設定した値は、スタンバイが解除されない場合、そのまま保持される。そして、設定変更をする周辺ユニットだけ、スタンバイ状態から解除して、新たな値を設定して、IDDQテストが可能となるという特徴がある。

【0044】これによって、IDDQテストのプログラム容量を減少させ、実行するテストボタンを短縮することはでき、テストコストを改善している。

【0045】さらに、IDDQテストプログラムの作成を容易化しており、製品開発期間、特に、IDDQテストプログラムの作成工数の短縮を図ることができる。

【0046】

【実施例】上記した本発明の実施の形態についてさらに詳細に説明すべく、本発明の実施例について図面を参照して以下に説明する。図1は、本発明の一実施例の構成を示す図である。図1を参照すると、本発明の一実施例は、CPU1のほかに、周辺ユニットとしてタイマ2、シリアルインタフェース3と、メモリ4を備え、さらに、テストモード設定レジスタ8を備えている。なお、図1に示す構成例では、周辺ユニットは、タイマ2、シリアルインタフェース3、メモリ4から構成されているが、マイクロコンピュータ内に設けられる周辺ユニットに種類、個数は任意である。

【0047】CPU1から出力されるスタンバイ信号10と、テストモード設定レジスタ8から出力される制御信号12とを入力し、スタンバイ制御信号14をタイマ2へ供給する第1のスタンバイイネーブル制御回路5と、スタンバイ信号10とテストモード設定レジスタ8から出力される制御信号11とを入力し、スタンバイ制御信号15をシリアルインタフェース3へ供給する第1のスタンバイイネーブル制御回路6と、スタンバイ信号10とテストモード設定レジスタ8から出力される制御信号13とを入力し、スタンバイ制御信号16をメモリ4へ供給する第3のスタンバイイネーブル制御回路7

と、を備え、CPU1、タイマ2、シリアルインタフェース3、メモリ4、テストモード設定レジスタ8は、内部バス9に接続されており、またタイマ2から出力されるタイミング信号17はシリアルインタフェース3へ入力されシリアル動作の基準クロックとなる。

【0048】第1～第3スタンバイネーブル制御回路5、6、7は、いずれも、テストモード設定レジスタ8から出力される制御信号と、スタンバイ信号10の論理ORをとる回路からなり、同一構成とされている。

【0049】図1に示した本発明の一実施例と、図4を参照して説明した従来のマイクロコンピュータとの相違点は、本発明の一実施例では、CPU1が所定の値を設定するテストモード設定レジスタ8と、各々の周辺ユニット2、3、4に対してそれぞれ付加されるスタンバイネーブル制御回路5、6、7を備えていることである。

【0050】スタンバイ状態からの解除を行わない周辺ユニットでは、付加したスタンバイネーブル制御回路から出力されるスタンバイ制御信号がアクティブ状態のままとなるように制御される。これにより、任意の周辺ユニットがスタンバイ状態を維持できるようにしている。

【0051】次に本発明の一実施例の動作について説明する。図2は、本発明の一実施例のマイクロコンピュータのIDDQテスト測定の流れを示すフローチャートである。図1及び図2を参照して、本発明の一実施例の動作を説明する。

【0052】初めにIDDQテストの測定条件をメモリ4で2条件、タイマ2で2条件、シリアルインタフェース3で2条件とする。IDDQテストは、タイマ2からシリアルインタフェース3へ接続するタイミング信号17があるので、タイマ2とシリアルインタフェース3は組み合わせ条件がは、

- ・タイマ条件1－シリアル条件1、
- ・タイマ条件1－シリアル条件2、
- ・タイマ条件2－シリアル条件1、
- ・タイマ条件2－シリアル条件2、

の4通りとなる。

【0053】ここで、

- ・タイマ条件1は、タイミング信号17がアクティブの条件、
  - ・タイマ条件2はタイミング信号17がインアクティブの条件、
- である。

【0054】CPU1はタイマ2に条件1、シリアルインタフェース3に条件1、メモリ4に条件1を設定する(ステップ201～203)。次にCPUはスタンバイ命令を実行し(ステップ204)、スタンバイ信号10がアクティブとなり、スタンバイ状態に移行する。その時のスタンバイ電流を測定し(ステップ205)、正常

であるか判断する。この時、タイマ2から出力されるタイミング信号17はアクティブ条件でシリアルインタフェース3へ入力される。

【0055】スタンバイ状態が解除される時、CPU1はテストモード設定レジスタ8へメモリ4のみスタンバイ状態を解除する値を書き込んで、スタンバイ状態から通常動作に切り替える。このため、メモリ4のみスタンバイ状態が解除される(ステップ206)。

【0056】次に、CPUはメモリ4へ条件2を設定し(ステップ207)、スタンバイ命令を実行してスタンバイ状態へ移行する(ステップ208)。この時も同様にスタンバイ電流を測定し、正常であるか判断する(ステップ209)。

【0057】スタンバイ状態が解除される時、CPU1はテストモード設定レジスタ8へシリアルインタフェース3のみスタンバイ状態を解除する値を書き込んで、スタンバイ状態から通常動作に切り替える。このため、シリアル4のみスタンバイ状態が解除される(ステップ210)。

【0058】次に、CPUはシリアルインタフェース3へ条件2を設定し(ステップ211)、スタンバイ命令を実行して(ステップ212)、スタンバイ状態へ移行する。この時も同様にスタンバイ電流を測定し(ステップ213)、正常であるか判断する。

【0059】スタンバイ状態が解除される時、CPU1はテストモード設定レジスタ8へタイマ2のみスタンバイ状態を解除する値を書き込んで、スタンバイ状態から通常動作に切り替える。このため、タイマ2のみスタンバイ状態が解除される(ステップ214)。

【0060】次に、CPUはタイマ2へ条件2を設定し(ステップ215)、スタンバイ命令を実行して(ステップ216)、スタンバイ状態へ移行する。この時も同様にスタンバイ電流を測定し(ステップ217)、正常であるか判断する。この時、タイマ2から出力されるタイミング信号17はインアクティブ条件となりシリアルインタフェース3へ入力される。

【0061】スタンバイ状態が解除される時、CPU1はテストモード設定レジスタ8へシリアルインタフェース3のみスタンバイ状態を解除する値を書き込んで、スタンバイ状態から通常動作に切り替える。このため、シリアルインタフェース3のみスタンバイ状態が解除される(ステップ218)。

【0062】次に、CPUはシリアルインタフェース3へ条件1を設定し(ステップ219)、スタンバイ命令を実行して(ステップ220)、スタンバイ状態へ移行する。この時も同様にスタンバイ電流を測定し(ステップ221)、正常であるか判断する。

【0063】スタンバイ状態が解除される時、CPU1はテストモード設定レジスタ8へ全ての周辺ユニットがスタンバイ状態を解除する値を書き込んで、全ての周辺

ユニットのスタンバイ状態が解除され（ステップ 22 2）、通常動作に復帰し、IDDQテストを終了する。

【0064】このIDDQテストでは、合計5回のスタンバイ電流が測定され、マイクロコンピュータが正常であるか判断される。本発明の第1の実施例では、22の処理ステップよりなり、図5に示したIDDQテストシーケンスと比べて、8ステップも処理を短縮している。

【0065】次に、本発明の第2の実施例について以下に説明する。図3は、本発明のマイクロコンピュータのスタンバイ制御装置の第2の実施例の構成を示す図である。

【0066】図3を参照すると、本発明の第2の実施例は、CPU1、タイマ2、シリアルインタフェース3、メモリ4と、テストモード設定レジスタ8を備えて構成されている。図3に示す例では、周辺ユニットは、タイマ2、シリアルインタフェース3、メモリ4だけから構成されているが、本発明において、周辺ユニットの種類、個数はかかる構成に限定されるものでないことは勿論である。

【0067】CPU1から出力されるスタンバイ信号10と、テストモード設定レジスタ8から出力される制御信号12とを入力し、スタンバイ制御信号14をタイマ2へ供給する第1のスタンバイイネーブル制御回路5と、スタンバイ信号10とテストモード設定レジスタ8から出力される制御信号11とを入力し、スタンバイ制御信号15をシリアルインタフェース3へ供給する第1のスタンバイイネーブル制御回路6と、スタンバイ信号10とテストモード設定レジスタ8から出力される制御信号13とを入力し、スタンバイ制御信号16をメモリ4に供給する第3のスタンバイイネーブル制御回路7と、を備え、CPU1、タイマ2、シリアルインタフェース3、メモリ4、テストモード設定レジスタ8は、内部バス9に接続されており、またタイマ2から出力されるタイミング信号17はシリアルインタフェース3へ入力されシリアル動作の基準クロックとなる。

【0068】前記第1の実施例では、テストモード設定レジスタ8へは、CPU1から内部バス9を介して制御データが設定されていたが、本発明の第2の実施例においては、テストモード設定レジスタ8へは、外部から入力される制御信号18、19、20が入力されている。

【0069】第1～第3スタンバイイネーブル制御回路5、6、7は、いずれも、テストモード設定レジスタ8から出力される制御信号と、スタンバイ信号10の論理ORをとる回路からなり、同一構成とされている。

【0070】本発明の第2の実施例においては、CPU1がプログラム命令でテストモード設定レジスタ8にスタンバイ制御情報を書き込むことは不要とされており、IDDQテストプログラムのサイズを短縮することができる。

【0071】テストモード設定レジスタ8へ入力される

制御信号18、19、20は、テスト時に、LSIテストより、テストタイミングに応じて、固定値を与えれば良い。テストモード設定レジスタ8へ入力される制御信号は、テストのときのみ使用するもので、他の機能を持つ汎用ポートと兼用することも可能である。

【0072】また、前記実施例では、テストモード設定レジスタ8の構成を例に本発明を説明したが、このテストモード設定レジスタ8の構成に限らず、スタンバイ状態を維持するか解除するかを選択が、LSIのIDDQテストの時に選択できる手段であれば、本発明を実施することができる。

【0073】

【発明の効果】以上説明したように、本発明のマイクロコンピュータのスタンバイ制御装置によれば下記記載の効果を奏する。

【0074】本発明の第1の効果は、テストプログラムの開発を簡易化し、かつ開発工数を削減し、開発期間を短縮することができる、ということである。

【0075】その理由は、本発明においては、内蔵する周辺ユニットがスタンバイ解除でスタンバイ状態から抜けるかどうかをテストのとき制御できるように構成したことにより、周辺ユニットのスタンバイ状態を任意の設定で維持することが可能となり、テストプログラムでは状態を変化させたい周辺ユニットの設定を変更するだけでよいからである。

【0076】本発明の第2の効果は、スタンバイ状態に設定するに要するプログラム量を大幅に削減することができ、この結果、テスト時間を短縮し、マイクロコンピュータのテストコストを低減する、ということである。

【0077】その理由は、本発明においては、内蔵する周辺ユニット全てにスタンバイ状態での条件設定をする必要が無く、変更したい周辺ユニットのみ条件を設定することができるため、である。

【0078】さらに、本発明の第3の効果は、安価なLSIテストでもテストを行うことができる、ということである。

【0079】その理由は、本発明においては、IDDQテストパタンの容量も縮減できるためである。

【図面の簡単な説明】

【図1】本発明の第1の実施例の構成を示す図である。

【図2】本発明の第1の実施例におけるIDDQテストのプログラムの処理手順を示す流れ図である。

【図3】本発明の第2の実施例の構成を示す図である。

【図4】従来のスタンバイ機能を備えたマイクロコンピュータの構成を示す図である。

【図5】従来のスタンバイ機能を備えたマイクロコンピュータにおけるIDDQテストのプログラムの処理手順を示す流れ図である。

【図6】特開平10-283275号公報に記載される集積回路の構成を示す図である。



13

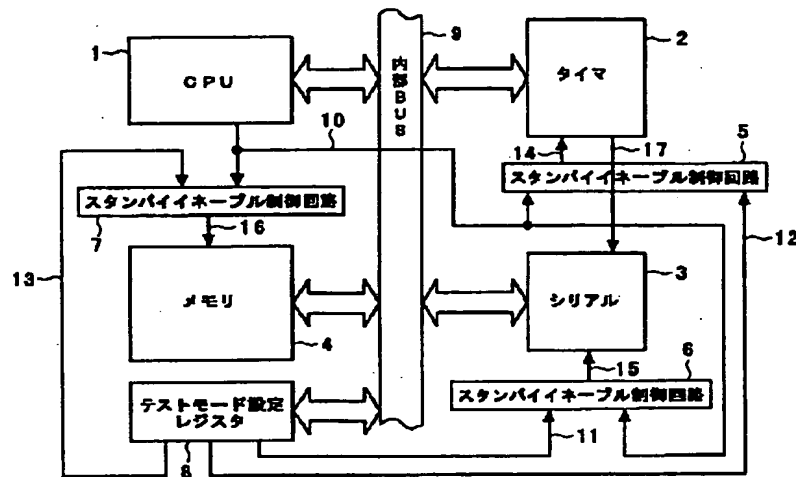
14

## 【符号の説明】

- 1 CPU  
2 タイマ  
3 シリアルインタフェース  
4 メモリ

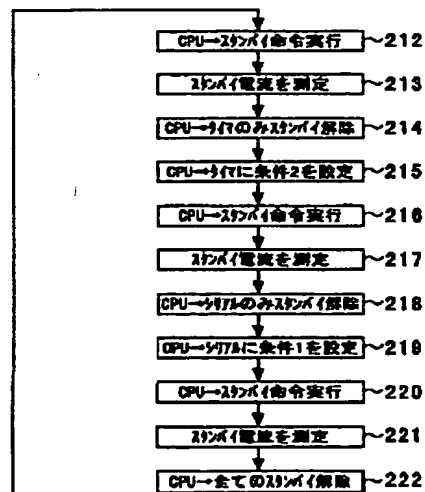
- 5、6、7 スタンバイネーブル制御回路  
8 テストモード設定信号  
9 内部バス  
10 スタンバイ信号

【図1】

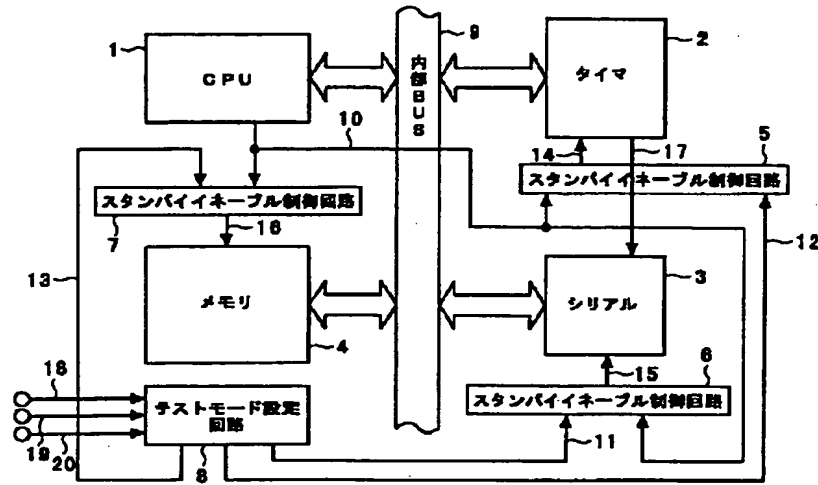


【図2】

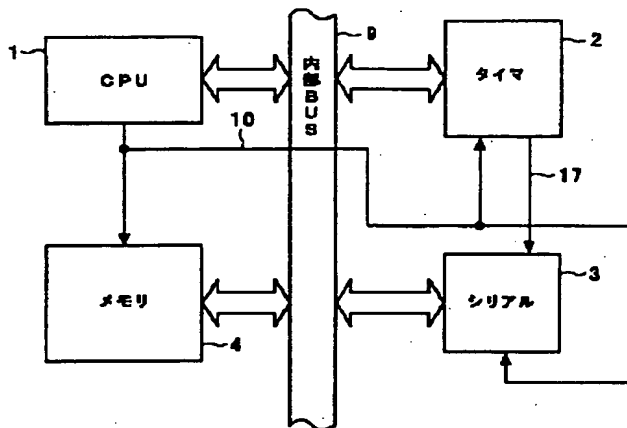
## 1000番プログラムの流れ



【図3】



【図4】

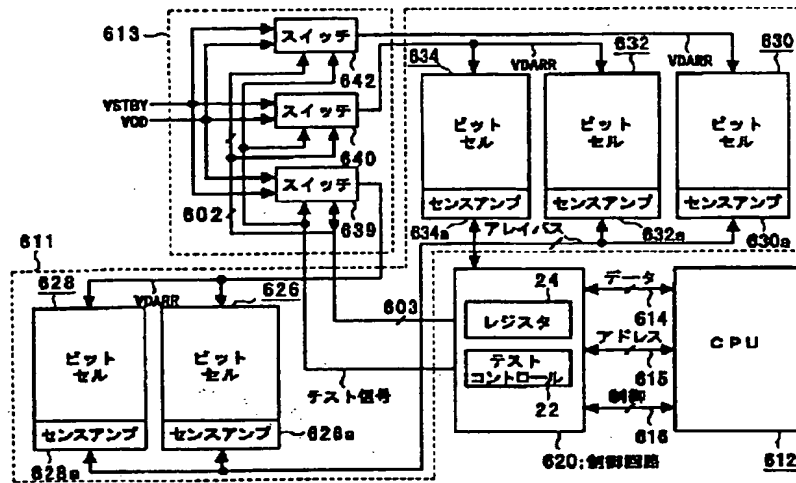


【図5】

1000測定プログラムの流れ



【図6】



フロントページの続き

Fターム(参考) 2G003 AA08 AB01 AB02 AE01 AF06  
 AH01 AH02 AH04 AH10  
 2G032 AA07 AB01 AC03 AD01 AE10  
 AE12 AG07 AK14  
 5B048 AA11 AA20 EE01 FF01  
 5B062 AA05 CC01 HH06 JJ05  
 9A001 BB05 LL05